1- Què és un framework?. Per què s'utilitzen frameworks?

Un framework PHP proporciona biblioteques de codi que poden ser utilitzades pel desenvolupador reduint d'aquesta manera la quantitat de codi original que el desenvolupador necessitarà escriure.

Per què utilitzar un framework?:

- Reduir el temps que cal per desenvolupar una aplicació.
- Reduir la quantitat de codi original a escriure.
- Utilització de codi ampliament provat i molt eficient.
- Obliga a seguir bones pautes a l'hora de desenvolupar una aplicació. Per exemple, el framework de Laravel obliga a utilitzar el mètode MVC de programació.
- Millora la seguretat de l'aplicació.
- Permet treballar en equip de manera més eficient.
- Manteniment del codi més senzill.

2- Programa gestor de dependències per treballar amb frameworks: composer

a) El programa **composer** és una eina per a la gestió de dependències en PHP. Us permet declarar les biblioteques de les quals depèn el vostre projecte i les gestionarà (instal·larà/actualitzarà).

b) El programa **composer** no és un gestor de paquets en el mateix sentit que aptitude. Sí, s'ocupa de "paquets" o biblioteques, però els gestiona per projecte, instal·lant-los en un directori que habitualment té el nom **vendor**(proveïdor) dins del vostre projecte.

c) Per defecte, no instal·la res globalment, només pel projecte. Per tant, és un gestor de dependències que només afecta al projecte amb el qual treballem. Tanmateix té opcions per fer disponibles dependències de manera global a meś d'un projecte.

d) Suposem que tens un projecte que depèn de diverses biblioteques. Algunes d'aquestes biblioteques depenen d'altres biblioteques. El programa **composer**:

- Permet declarar les biblioteques de les quals el projecte en depén
- Esbrina quines versions de quins paquets es poden i cal instal·lar i les instal·la (és a dir, les baixa al vostre projecte).
- Actualitza totes les dependències de les quals depenen les biblioteques del projecte.

e) Quan afegim un framework a un projecte, dins de la carpeta del projecte es crearan (o actualitzaran) dos fitxers:

- **composer.json** → Llistat a on s'especifica les necessitats del projecte, que bàsicament vol dir que s'indica quines són les seves dependències del projecte, restriccions de versions de les dependències, restriccions de versió de PHP i extensions requerides de PHP. L'ordre **`composer`** instal·larà les versions demanades dins de les dependències o una de mes recent.
- **composer.lock** → Llistat de les versions exactes de les dependències instal·lades i altres dades esencials per poder actualitzar dependències amb l'ordre **`composer update`** o realitzar noves instal·lacions del projecte a partir de dependències actualitzades amb **`composer install.`**

f) Instal·lació:

- Debian Linux -->

```
sudo aptitude update
sudo aptitude install php-xml
sudo aptitude install composer
```

- Windows --> https://getcomposer.org/doc/00-intro.md#installation-windows

g) Documentació: https://getcomposer.org/doc/00-intro.md

4- Planificació

a) És convenient tenir la precaució de planificar els frameworks que ens faran falta abans de començar el projecte i afegir-os a la carpeta del projecte abans de començar a afegir el nostre codi.

b) Sempre es pot afegir un framework al projecte un cop començat però millor que ja s'hagin afegit i estiguin disponibles des del principi.

5- Framework Dompdf

a) **Dompdf** és un **framework** per convertir fitxers **html** a **PDF** i mostrar-lo per mitjà del programa client que habitualment serà un navegador.

b) Requeriments bàsics: **PHP v7.1** o superior.

c) Si ho demana (però en funció de la versió, no caldrà) s'haura d'habilitar l'utilització de l'extensió del mòdul **MBSTRING** de **PHP**:

- **Linux:**
  - Modifica del fitxer **/etc/php/8.4/apache2/php.ini**. Descomenta **extension=mbstring**
  - Reinicia **apache2** → **sudo systemctl restart apache2**
  - **NOTA:** Si vols també poder executar el script des del terminal, també et cal descomentar **extension_mbstring** del fitxer **/etc/php/8.4/cli/php.ini**.

- **Windows (XAMPP):**
  - Modifica del fitxer **c:\xampp\php\php.ini**. Descomenta **extension=php_mbstring.dll**
  - Reinicia **apache2** des del taulell de control de **XAMPP**.

d) Instal·lació dins d'un projecte:

- Crea la carpeta del projecte
- Executa dins de la carpeta del projecte: **`composer require dompdf/dompdf`**
- Comprova que s'ha creat la carpeta **vendor** dins de la carpeta de projecte.
- Comprova que s'ha creat la carpeta **dompdf** dins **vendor**.
- Comprova que s'ha creat el fitxer **composer.json** dins de la carpeta de projecte.
- Comprova que s'ha creat el fitxer **composer.lock** dins de la carpeta de projecte.

e) Carpetes i fitxers dins del projecte:

- **vendor** → Carpeta a on es desen els frameworks que utilitza el projecte.
- **vendor/dompdf** → Carpeta del framework **Dompdf**.
- **composer.json** → Llistat a on s'especifica les dependències del projecte, és a dir, a on s'escriu la llista dels frameworks que necessita el projecte indicant els seus noms i la versió requerida. L'ordre **`composer`** instal·larà la versió demanada o una de mes recent.
- **composer.lock** → Llistat de les versions exactes de les dependències instal·lades i altres dades esencials per poder actualitzar dependències amb l'ordre **`composer update`** o realitzar noves instal·lacions del projecte a partir de dependències actualitzades amb **`composer install`.**

e) Utilització dins d'una aplicació PHP:

```
<?php
   require_once 'vendor/autoload.php';
   use Dompdf/Dompdf;
   $dompdf = new Dompdf;
   ....
```

f) S'ha d'habilitar l'utilització de l'extensió del mòdul MBSTRING de PHP

g) Mètodes de **Dompdf** --> https://gennco.com.co/ANT/dompdf/DOMPDF.html

h) Codi font i documentació: https://github.com/dompdf/dompdf

i) Exemple d'utilització: https://github.com/dacomo2122daw2/dacomo2122daw2-html2pdf.git